Experimenting the 1149.1 and 1149.4 test infrastructures in a Web-accessible remote Lab (without Plug-ins!)

André Vaz Fidalgo[1,2] avf@dee.isep.ipp.pt
Ricardo J. Costa[2] rjcosta@fe.up.pt
Gustavo R. Alves[1,2] galves@dee.isep.ipp.pt
José M. Ferreira[2] jmf@fe.up.pt

[1]Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida 431
P-4200-072 PORTO

[2]Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias
P-4200-465 PORTO

Abstract

The expansion of the Internet has supported the development of online teaching resources based on this communication media (e-learning). However, the possibility to run experiments on remote-accessible Labs, in a teaching context, is a more recent fact. This paper describes the framework for delivering through the Web, a course on Design for Debug and Test that contains several practical exercises involving the use of the IEEE 1149.1 and 1149.4 test infrastructures. The exercises are done in a remote-accessible Lab, installed at our facilities, through a simple interface readable on any web browser. By using a complete Java-based solution, there is no need for installing any sort of plug-ins at the client computer as it happens in other similar approaches that require some sort of downloading extra software or the development of Common Gateway Interfaces (CGIs).

1. Introduction

The work described in this paper is a result of previous and on-going projects:

- INSIGHT II (1996 to 1999) [1]
- ASTEP (Mar 1997 – Feb 2000) [2]
- ALLEGRO (Jan 2000 – Jul 2001) [3]
- PEARL (Mar 2000 – Feb 2003) [4]

The Leonardo INSIGHT II project provided the opportunity to develop classic course materials (lecture notes + overheads) on modern electronic systems design, which included a complete workpackage on “Development and Test of Digital and Mixed-signal Systems”. This has been the main working area of the authors for the past ten years, with a special focus on the IEEE 1149.x suite of standards [5, 6, 7, 8]. The ASTEP project enabled the Web-based delivery of a select portion of the materials developed under INSIGHT II. Pitfalls pointed out at the end of ASTEP have been addressed by the on-going ALLEGRO and PEARL projects. The first includes the quality control and pedagogical aspects of Internet / Intranet course management and delivery. These pedagogical aspects are observed in the passage of course materials on Design for Debug and Test (developed during the INSIGHT II project) to a Web-based format, which is reflected in the definition of a sound assessment methodology. The second addresses the development of an Internet-accessible Lab infrastructure for remote experimentation, the main idea being that if a course is delivered though the Internet then the hands-on materials that support the practical exercises (one of the three components of the course assessment scheme) should also be made available through the same media (and not delivered as course material boxes sent to each and every student). This idea has been pursued in the past [9], although only more recent solutions have dispensed the use of plug-ins [10, 11].

The following sections describe in more detail these components (the web-based course delivery system and its main features, the course development process and the associated assessment methodology, and the web-accessible lab supporting the remote experimentation), leading the way to the presentation of one actual exercise involving the IEEE 1149.1 test infrastructure [12].

2. Course contents, assessment methodology and delivery system

The selected course delivery system was the well-known WebCT®, which is one of the most widely accepted solutions in this domain. WebCT® allows the management of student databases, HTML content modules and several assessment/monitoring and communication tools consisting of a restricted e-mail system, a multi-subject message forum and a real-time chat environment. The assessment methodology defined for this course in consistence with the principles established in [13], comprehends three main components:

- Self-tests associated to each content page
- Assignments presented at the end of each chapter
- A final test at the end of the course

Each self-test consists of several multiple-choice questions, where the student can get immediate feedback in the form of comments to each answer. This component allows the student to evaluate its own knowledge without being actually graded (formative assessment). On the other hand, the final test is only made available on a pre-determined time frame and its results count for the final grade (summative assessment). The user can chose the time and place to answer the test, with WebCT® being responsible for the control of the test time, as well as the management of the questions posed to each student. These are randomly chosen from a database that covers the entire course content.
The assignments are proposed to the students not only as an evaluation tool but also as a learning method. After each section is fully studied, the student is given the opportunity to use its new knowledge on a practical exercise. These are presented with some diversity, including theoretical studies, research, and design problems. A very particular type of assignments proposed in the course follow the principle of remote experimentation. To this effect we use a Lab infrastructure connected to the Internet, in the form of a PXI-based instrumentation system. As the experiment interface is presented via a web browser, it becomes remotely accessible, being even possible for the user to actually view the results of its actions through a WebCam. The principal advantage of this approach is the sharing of centralized resources, i.e. it is not necessary to supply each student with the material to perform each experiment. Figure 1 illustrates several aspects of the course: a page presenting one assignment (left), a content page (right) with an enhanced view of the tools bar (top) containing links to the WebCT® communications tools, and the associated self-test.

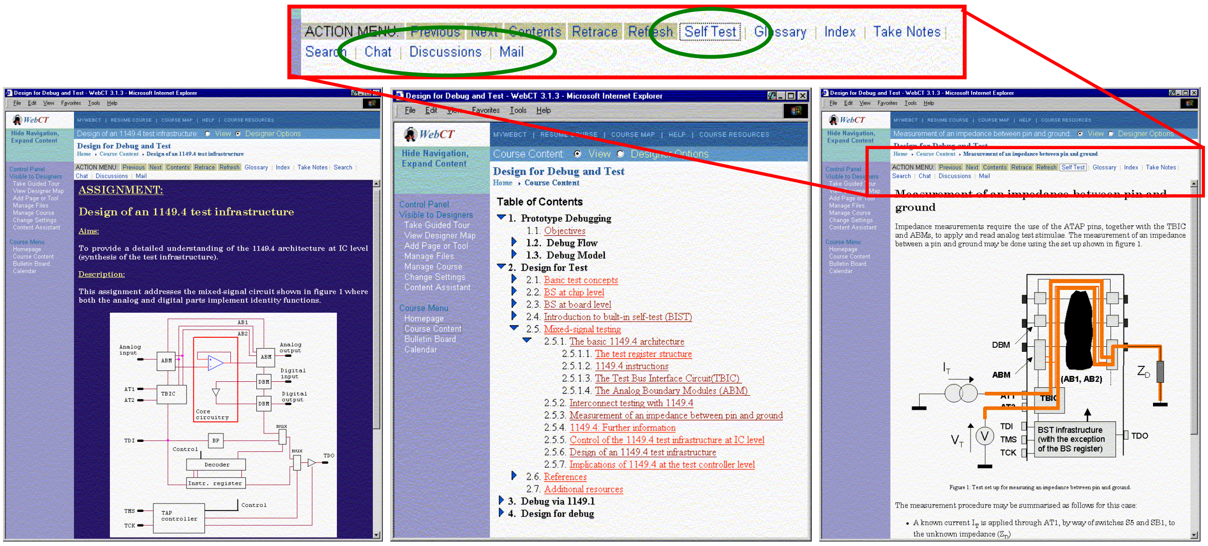

Figure 1: A composed view of several course HTML pages.

The HTML page illustrated in the centre of figure 1 presents the table of contents of the course, which is divided in four main chapters:

- Prototype debugging – presents the basic concepts on prototype debugging tools, techniques & operations
- Design for Test(ability) – presents the basic test concepts, the IEEE 1149.1 and 1149.4 test infrastructures, its implementation and use at the chip and board level, and test protocols.
- Debug via 1149.1 – presents the implementation of debug operations using the mandatory/optional instructions described in the standard, plus other optional instructions available in components from the SCOPE™ and SCAN™ families.
- Design for Debug – describes the outlining and implementation of new user-defined optional instructions that cover the pitfalls identified in the previous chapter, mainly in the area of detecting breakpoints in real-time and real-time acquisition.

The remote lab provides support for the assignments proposed in the second and third chapters that address the presentation and use of the 1149.1 and 1149.4 test infrastructures for debug and test purposes. These two aspects (the remote lab apparatus and interface, and the actual experiments) are described in the following sections.

3. The remote Lab infrastructure

The remote lab comprehends several hardware and software modules that provide control / monitor privileges to the Unit Under Test (UUT) to any user with a PC connected to the Internet, as illustrated in figure 2. The UUT consists of a board with several digital and analogue components that all together implement a basic 1149.4 test infrastructure. The board is connected to several instrumentation slots (Oscilloscope, Function Generator, Switch-Matrix, and Boundary Scan Controller) that allow its control and monitoring. Users with access to this kind of instruments have the facility to interact with the UUT, making configurations and catching results, like if they were in the Lab. In order to emphasize the reality of the experience, we installed a WebCam that allows students to see results of the actions and the laboratory where all takes place.

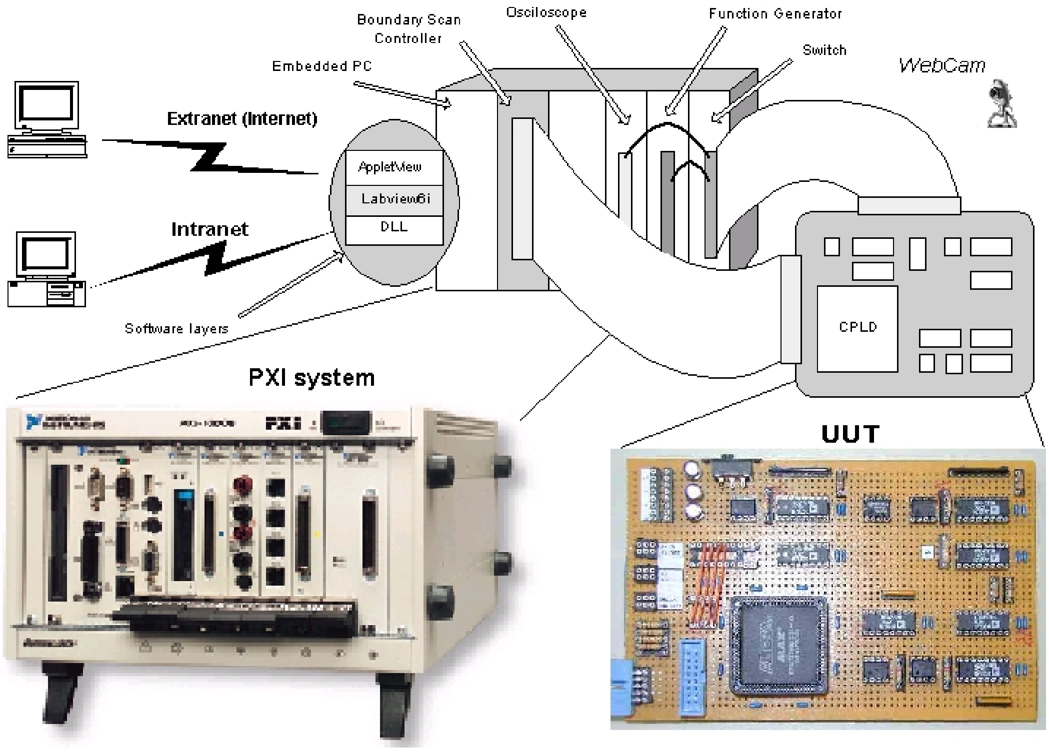

Figure 2: A conceptual view of the remote lab with real pictures of the hardware components.

The instruments are connected through a PXI ("PCI eXtended Instrumentation") bus, which includes an embedded PC that runs all the software. The instrument interfaces are developed with LabView, a general-purpose programming system that includes libraries of functions and development tools designed specifically for data acquisition and instrumentation control. LabView programs are named VIs (Virtual Instruments) because their appearance and operation imitate actual instruments. A VI consists of an interactive user interface and a dataflow diagram corresponding to the source code, as illustrated in figure 3.

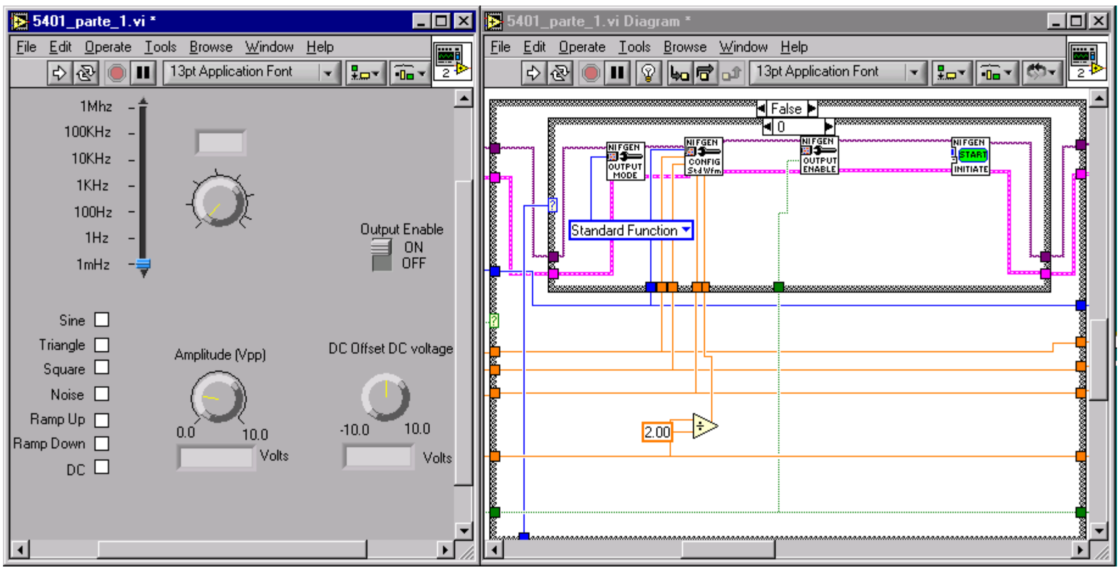

Figure 3: Example of a VI formed by its interactive user interface and dataflow diagram.

To make the experiments accessible through the Web we use the AppletView software that allows the constructions of Applets. These are in reality GUIs ("Graphical Units Interfaces") that are downloaded in the user browser each time there is an access to the remote experiment. AppletView is a software kit, from Nacimiento, for creating interactive Java Applets that communicate in real-time with VIs. It allows a user to take advantage of existing TCP/IP networks and Java-enabled Web Browsers (such as Netsacpe Communicator or Internet Explorer) to access an instrumentation system. One important feature of this software is its compatibility with any system (it is entirely developed in Java) thus dispensing the download or use of any Plug-Ins. At the present moment this software is still under development, with ourselves acting as Beta-users. The current Beta version 4.0 still has some limitations that are posing problems in the creation of the interfaces for the oscilloscope, the switch-matrix, and the function generator. Following close contacts with this company, we expect to see these problems overcome in the next Beta version.

AppletView is a central component of our remote Lab, as it is responsible not only for creating the Java version of the VI but also for acting as the server that handles all user requests. This software must be running on the remote Lab server (in our case the PC embedded in the PXI system) to enable exterior access to each experiment, as expressed in figure 4. The result of processing a VI with this software is illustrated in figure 5, which emphasizes the top bars showing the identification of the VI (Tapper.vi – on the top left hand) and of the URL of the Java-version accessible through the Web (http://localhost:8080/viml/tapper.viml - on the top right hand).

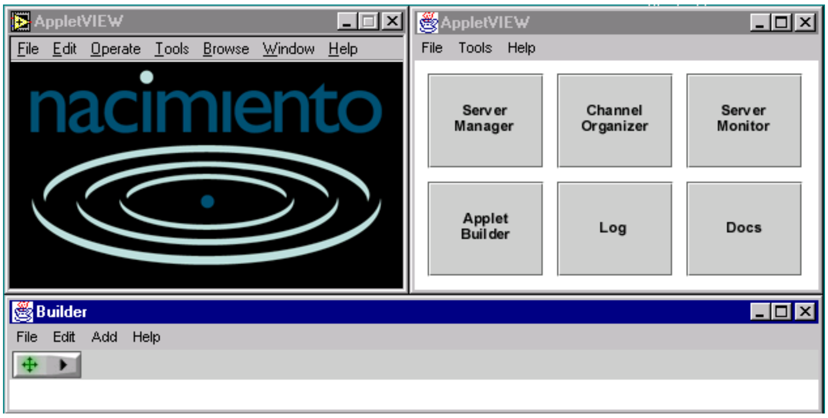

Figure 4: Print-Screen of AppletView acting as the Web server (top right) after creating the Java-based interface of a given VI (bottom).

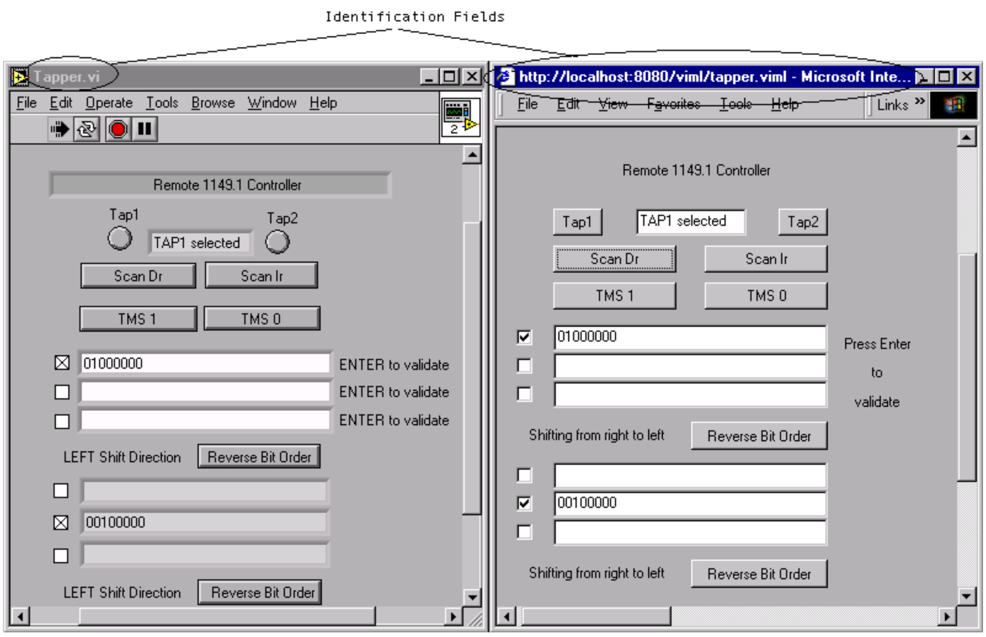

Figure 5: Example of a VI and the corresponding Java-based interface that allows the remote control of up to two Boundary Scan chains.

4. Example of an experimentation session with the 1149.1 test infrastructure

Some of the assignments included along the course address the IEEE 1149.1 and 1149.4 test infrastructures. The goal of such assignments is to provide a basic understanding of the form and use of these architectures. The interface illustrated in figure 5 allows a remote user to control, in real-time, up to two Boundary Scan (BS) chains of any board connected to the PXI-system (that includes a BS controller from Göpel) [14]. One of the first assignments consists of asking the user to control, through the associated BS register, the state of the eight output pins of a 74BCT8244 (an unidirectional non-inverting buffer with eight channels, from the SCOPE™ family, Texas Instruments). The state of these outputs is detectable through eight LEDs that, in turn, are visible through the WebCam, as illustrated in figure 6.

It is not our intent to describe the protocol necessary for placing a certain logic value in one or more pins, through the corresponding BS cells, but in any case it is possible to say that the purpose of the assignment is to actually check if the student has well understood that protocol, explained in the course content pages that precede the one presenting the exercise. The point is that by having a simple and powerful BS controller interface, made accessible through the Web, and by getting an immediate feedback from the WebCam [15], the student is encouraged to practice its recently acquired knowledge on the 1149.1 test infrastructure.

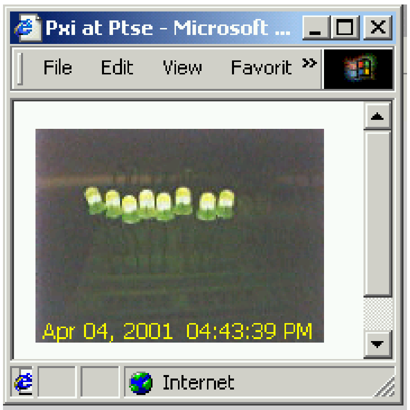

Figure 6: Image from the WebCam.

5. Conclusions

This paper described the framework for delivering, through the Web, a course on Design for Debug and Test (based on the 1149.1 and 1149.4 test infrastructures). Practical or hands-on exercises, which are an essential part of any course on electronics, may be done in a remote-accessible Lab. By using a complete Java-based access scheme, there is no need for installing plug-ins at the client computer as it happens in other similar previous works [9], or the development of CGIs that require an additional level of expertise since LabView is also used for developing the VIs [10,11]. Immediate feedback is provided to the user in form of results sent by the remote instruments and by a WebCam. Course trial sessions have already been made in Norway and Portugal, while others are planned to occur in the near future in Scotland and Lithuania.

References

[1] INSIGHT II, project N/96/2/1085/PI/II.1.1.c/FPC, http://kong.hibu.no/avdeling/data/insight/

[2] ASTEP, project MM1001, http://www.cee.hw.ac.uk/~astep/

[3] ALLEGRO, project P/99/1/075270/PI/III.3.a/FPC, http://www.fe.up.pt/~allegro/index.htm

[4] PEARL, project IST 1999/12550 http://kmi.open.ac.uk/projects/pearl/index.htm

[5] J. S. Matos, F. S. Pinto and J. M. Ferreira, "*A Boundary Scan Test Controller for Hierarchical BIST*," International Test Conference (ITC'92), Baltimore, USA, September 1992.

[6] Jose M. Ferreira, M. G. Gericota, J. L. Ramalho and Gustavo R. Alves, "*BIST for 1149.1 - Compatible Boards: A Low-cost and Maximum Flexibility Solution*", International Test Conference (ITC'93), Baltimore, USA, October 1993.

[7] Gustavo R. Alves and J. M. Ferreira, "*From Design-for-Test to Design-for-Debug-and-Test: Analysis of Requirements and Limitations for 1149.1*", VLSI Test Symposium (VTS), Dana Point, USA, April 1999.

[8] Gustavo R. Alves, Margrit Reni Krug, Marcelo Lubaszewski and Jose M. Ferreira, "*From Circuit Simulation to Circuit Verification: An Internal + Boundary Scan-based Solution*", European Test Workshop (ETW'00), Cascais, Portugal, May 2000.

[9] Marcin A. Stegawski and Rolf Schaumann, "*A New Virtual-Instrumentation-Based Experimenting Environment for Undergraduate Laboratories with Application in Research and Manufacturing*", IEEE Transactions on Instrumentation and Measurement. Vol. 47, N. 6, December 1998.

[10] Pasquale Arpaia, Aldo Baccigalupi, Felice Cennamo, and Pasquale Daponte, "*A Measurement Laboratory on Geographic Network for Remote Test Experiments*", IEEE Transactions on Instrumentation and Measurement. Vol. 49, N. 5, October 2000.

[11] M. A. Allende, R. Mozuelos, M. Martínez, S. Bracho, "On line IC test course with distance access to test equipment", IEEE Latin American Test Workshop (LATW'01), February 2001.

[12] IEEE Standard Test Access Port and Boundary-Scan Architecture, Oct. 1993, IEEE Std. 1149.1 (Includes IEEE Std. 1149.1a), ISBN 1-55937-350-4.

[13] Dale Luksaite, "*Quality Control and Good Practises for Internet / Intranet Course Development and Delivery*", Workpackage 4, ALLEGRO project, June 2000.

[14] http://brama.fe.up.pt:8080/viml/tapper.viml

[15] http://pandora.fe.up.pt/webcam/webcam.htm